FORM PTO-1390 (REV 5-93) U S DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER: 2202/49999

U S APPLICATION NO (if known, see 37 CFR 1 5): 09/856615

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/FR99/02883	23 November 1999	24 November 1998

TITLE OF INVENTION
A DEVICE FOR AND A METHOD OF DETECTING STACK OVERFLOW IN A MEMORY AND A FRANKING MACHINE EMPLOYING THEM

APPLICANT(S) FOR DO/EO/US
Jean-Marc DERY and Frederic L'HOTE

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. [X] This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.

2. [ ] This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371

3. [X] This express request to begin national examination procedures (35 U.S.C. 371(f) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).

4. [X] A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.

5. [X] A copy of the International Application as filed (35 U.S.C. 371(c)(2)).
   a. [X] is transmitted herewith (required only if not transmitted by the International Bureau).
   b. [ ] has been transmitted by the International Bureau
   c. [ ] is not required, as the application was filed in the United States Receiving Office (RO/US)

6. [X] A translation of the International Application into English (35 U.S.C. 371(c)(2)).

7. [X] Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. [ ] are transmitted herewith (required only if not transmitted by the International Bureau).
   b. [ ] have been transmitted by the International Bureau.
   c. [ ] have not been made; however, the time limit for making such amendments has NOT expired.
   d. [X] have not been made and will not be made.

8. [ ] A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).

9. [X] An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)) (executed)

10. [ ] A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Item 11. to 16. below concern other document(s) or information included:

11. [X] An Information Disclosure Statement under 37 CFR 1.97 and 1.98.

12. [X] An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.

13. [X] A FIRST preliminary amendment.
    [ ] A SECOND or SUBSEQUENT preliminary amendment.

14. [ ] A substitute specification.

15. [ ] A change of power of attorney and/or address letter.

16. [X] Other items or information:
    a. 4 sheets of drawings showing Fig. 1-4;
    b. International Search Report.

23911
PATENT TRADEMARK OFFICE

U.S. APPLICATION NO (if known, see 37 CFR 1.5) 09/856615	INTERNATIONAL APPLICATION NO PCT/FR99/02883	ATTORNEY'S DOCKET NUMBER 2202/49999

17. [X] The following fees are submitted: Basic National Fee (37 CFR 1.492(a)(1)-(5)):	CALCULATIONS	PTO USE ONLY
Search Report has been prepared by the EPO or JPO ..... $860.00 International preliminary examination fee paid to USPTO (37 CFR 1.482) ... $690.00 No international preliminary examination fee paid to USPTO (37 CFR 1.482) but international search fee paid to USPTO (37 CFR 1.445(a)(2) ..... $710.00 Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37CFR 1.445(a)(2) paid to USPTO ..... $ 1000.00 International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2)-(4) $100.00		
ENTER APPROPRIATE BASIC FEE AMOUNT =	$860.00	
Surcharge of $130.00 for furnishing the oath or declaration later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(e)).	$	

Claims	Number Filed	Number Extra	Rate		
Total Claims	21- 20 =	1	X $18.00	$ 18.00	
Independent Claims	2 - 3 =	0	X $80.00	$	
Multiple dependent claims(s) (if applicable)			+ $270.00	$	
		TOTAL OF ABOVE CALCULATIONS =		$878.00	
Applicant claims Small Entity Status (See 37 CFR §1.27) [ ] yes [ ] no. Reduction by 1/2 for filing by small entity, if applicable.				$	
		SUBTOTAL =		$878.00	
Processing fee of $130.00 for furnishing the English translation later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(f)). +				$	
		TOTAL NATIONAL FEE =		$878.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28,3.31). $40.00 per property +				$ 40.00	
		TOTAL FEE ENCLOSED =		$918.00	
			Amount to be: refunded	$	
			charged	$	

a. [X] Two checks in the amount of $878.00 for the filing fee and $40.00 for the assignment recording fee are enclosed.
b. [ ] Please charge my Deposit Account No. _____ in the amount of $_____ to cover the above fees. A duplicate copy of this sheet is enclosed.
c. [X] The Commissioner is hereby authorized to charge any additional fees, which may be required, or credit any overpayment to Deposit Account No. 05-1323. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:
Crowell & Moring, L.L.P..
P.O. Box 14300
Washington, D.C. 20044-4300
Tel. No. (202) 628-8800
Fax No. (202) 628-8844

SIGNATURE
Jeffrey D. Sanok
NAME
32,169
REGISTRATION NUMBER
May 24, 2001
DATE

JDS:pct

Attorney Docket: 2202/49999
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: JEAN-MARC DERY et al.

Serial No.: NOT YET ASSIGNED PCT NO.: PCT/FR99/02883

Filed: MAY 24, 2001

Title: A DEVICE FOR AND A METHOD OF DETECTING STACK OVERFLOW IN A MEMORY AND A FRANKING MACHINE EMPLOYING THEM

PRELIMINARY AMENDMENT

Box PCT
Commissioner for Patents
Washington, D.C. 20231

Sir:

Please enter the following amendments to the specification, claims and abstract prior to the examination of the application.

IN THE SPECIFICATION:

Please amend the specification as follows:

Page 1, before the first full paragraph, insert the following heading:

--BACKGROUND AND SUMMARY OF THE INVENTION--.

Page 5, after the first full paragraph, insert the following heading:

--BRIEF DESCRIPTION OF THE DRAWINGS--.

Page 5, after the second full paragraph, insert the following heading:

--DETAILED DESCRIPTION OF THE DRAWINGS--.

IN THE CLAIMS:

Please amend claims 4-10 and 14-21 as follows:

(A copy of the marked-up version of amended claims are attached to this Preliminary Amendment).

4. (Amended) A method according to claim 1, characterized in that, during the allocation operation (402), the memory part (305) associated with a stack (304) is adjacent to it.

5. (Amended) A method according to claim 1, characterized in that, during the allocation operation (402), the memory part (305) associated with a stack (304) is adjacent to the next stack (306) in the order of writing the stack with which said memory part is associated.

6. (Amended) A method according to claim 1, characterized in that the reading and verification operations (407 to 409) are effected by a routine (405) of said program.

7. (Amended) A method according to claim 1, characterized in that said predetermined values are all the same.

8. (Amended) A method according to claim 1, characterized in that said predetermined values are different from all the values taken by computer codes of said program or from all values used in the stacks.

9. (Amended) A method according to claim 1, characterized in that when it is found, during the verification operation (409), that at least one read value has been modified, during a program modification operation (410), the execution of each program part relating to the stack associated with the memory part that has been read is suspended.

10. (Amended) A method according to claim 1, characterized in that when it is found, during the verification operation (409), that at least one read value has been modified, during a program modification operation (410), the execution of each program part relating to the stack which follows the stack associated with the read memory part in the order of writing the stacks is suspended.

14. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted to allocate to a stack (304) a memory part (305) which is adjacent to it.

15. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted to allocate to a stack (304) a memory part (305) which is adjacent to the stack (306) after it in the order of writing the stack with which said memory part is associated.

16. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted to read and verify values in the memory part by executing a routine of said program (405).

17. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted to process predetermined values that are all the same.

18. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted to process predetermined values that are different from all values taken by

computer codes of said program or from all values used in the stacks.

19. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted, if they determine that at least one read value has been modified, to modify the execution of the program, execution of each program part relating to the stack associated with the memory part that has been read being suspended (410).

20. (Amended) A device according to claim 11, characterized in that the processing means (106) are adapted, if they have determined that at least one read value has been modified, to modify the execution of the program, execution of each program part relating to the stack following the stack associated with the read memory part in the order of writing the stacks being suspended (410).

21. (Amended) A franking machine (1), characterized in that it includes a device according to claim 11.

IN THE ABSTRACT:

Please add an Abstract of the Disclosure submitted herewith on a separate page.

REMARKS

Entry of the amendments to the specification, claims and abstract before examination of the application is respectfully requested.

If there are any questions regarding this Preliminary Amendment or this application in general, a telephone call to the undersigned would be appreciated since this should expedite the prosecution of the application for all concerned.

It is respectfully requested that, if necessary to effect a timely response, this paper be considered as a Petition for an Extension of Time sufficient to effect a timely response and shortages in other fees, be charged, or any overpayment in fees be credited, to the Account of Crowell & Moring, L.L.P., Deposit Account No. 05-1323 (Docket #2202/49999).

Respectfully submitted,

May 24, 2001

Jeffrey D. Sanok
Registration No. 32,169

CROWELL & MORING, L.L.P.
P.O. Box 14300
Washington, DC 20044-4300
Telephone No.: (202) 628-8800
Facsimile No.: (202) 628-8844

JDS:pct

--<u>**ABSTRACT OF THE DISCLOSURE**</u>

The invention concerns a method for detecting overflow in at least one stack, a memory space reserved for part of a computer program. Said method consists in: for each monitored stack, assigning predetermined values to a memory part which, in said stack writing order, follows said stack; and for each implementation of a program part associated with said stack, reading the values present in said part of the memory and verifying the read values.--

VERSION WITH MARKINGS TO SHOW CHANGES MADE

Please amend claims 4-10 and 14-21 as follows:

4. (Amended) A method according to [any of claims 1 to 3] claim 1, characterized in that, during the allocation operation (402), the memory part (305) associated with a stack (304) is adjacent to it.

5. (Amended) A method according to [any of claims 1 to 4] claim 1, characterized in that, during the allocation operation (402), the memory part (305) associated with a stack (304) is adjacent to the next stack (306) in the order of writing the stack with which said memory part is associated.

6. (Amended) A method according to [any of claims 1 to 5] claim 1, characterized in that the reading and verification operations (407 to 409) are effected by a routine (405) of said program.

7. (Amended) A method according to [any of claims 1 to 6] claim 1, characterized in that said predetermined values are all the same.

8. (Amended) A method according to [any of claims 1 to 7] claim 1, characterized in that said predetermined values are different from all the values taken by computer codes of said program or from all values used in the stacks.

9. (Amended) A method according to [any of claims 1 to 8] claim 1, characterized in that when it is found, during the verification operation (409), that at least one read value has been modified, during a program modification operation (410), the execution of each program part relating to the stack associated with the memory part that has been read is suspended.

10. (Amended) A method according to [any of claims 1 to 9] claim 1, characterized in that when it is found, during the verification operation (409), that at least one read value has been modified, during a program modification operation (410), the execution of each program part relating to the stack which follows the stack associated with the read memory part in the order of writing the stacks is suspended.

14. (Amended) A device according to [any of claims 11 to 13] claim 11, characterized in that the processing means (106)

are adapted to allocate to a stack (304) a memory part (305) which is adjacent to it.

15. (Amended) A device according to [any of claims 11 to 14] claim 11, characterized in that the processing means (106) are adapted to allocate to a stack (304) a memory part (305) which is adjacent to the stack (306) after it in the order of writing the stack with which said memory part is associated.

16. (Amended) A device according to [any of claims 11 to 15] claim 11, characterized in that the processing means (106) are adapted to read and verify values in the memory part by executing a routine of said program (405).

17. (Amended) A device according to [any of claims 11 to 16] claim 11, characterized in that the processing means (106) are adapted to process predetermined values that are all the same.

18. (Amended) A device according to [any of claims 11 to 17] claim 11, characterized in that the processing means (106) are adapted to process predetermined values that are different

from all values taken by computer codes of said program or from all values used in the stacks.

19. (Amended) A device according to [any of claims 11 to 18] claim 11, characterized in that the processing means (106) are adapted, if they determine that at least one read value has been modified, to modify the execution of the program, execution of each program part relating to the stack associated with the memory part that has been read being suspended (410).

20. (Amended) A device according to [any of claims 11 to 19] claim 11, characterized in that the processing means (106) are adapted, if they have determined that at least one read value has been modified, to modify the execution of the program, execution of each program part relating to the stack following the stack associated with the read memory part in the order of writing the stacks being suspended (410).

21. (Amended) A franking machine (1), characterized in that it includes a device according to [any of claims 11 to 20] claim 11.

"A device for and a method of detecting stack overflow in a memory and a franking machine employing them"

The present invention relates to a device for and a method of detecting stack overflow in a memory and a franking machine employing them.

It applies in particular to franking machines provided with a program executing in a multitasking environment.

Correct execution of each task of a program must be guaranteed. Correct execution means that a task executes in its stack. The stack of a task corresponds to a memory space reserved for it. Thus, in other words, the invention aims to ensure that the memory used by a task is limited to the memory space allocated to it.

There is no certification of no stack overflow in prior art multitasking programs using electronic memories.

In the case of franking machines, the tasks can manage sums of money or data whose integrity must be assured. It is therefore essential to guarantee that no stack overflows beyond the memory space allocated to it.

To this end, the present invention aims to verify the integrity of the stack of the new task at each change of context within a program. The integrity of the stack is verified by checking the first bytes from the stack against a predefined value. If those bytes from the stack have an unexpected value, the stack is considered to have been violated and execution of at least one task of the program is stopped.

Thus, in accordance with the invention, a given number of bytes at the beginning of each stack are dedicated to overflow detection. Because all the stacks are in succession in a memory space, a multitasking kernel verifies the value of the bytes dedicated to overflow detection for each task that will execute. Accordingly, if

a stack overflows, the values of the bytes dedicated to detecting overflow of the next stack are modified. As soon as the multitasking kernel detects the modification, execution of at least one of the tasks affected by the stack that has overflowed, or by the stack to which the overflow occurs, is stopped.

To this end, on each change of context, a program routine is executed to verify the value of the bytes dedicated to overflow detection corresponding to the stack attached to the next task to execute.

A first aspect of the present invention provides a method of detecting overflow of at least one stack, i.e. of a memory space reserved for a part of a computer program, characterized in that it includes:

- for each stack monitored, an operation of allocating predetermined values to a memory part which is after said stack in the order of writing said stack, and
- on each execution of a program part associated with said stack, an operation to read values present in said memory part and an operation to verify the read values.

Accordingly, if the stack to which a memory part has been allocated overflows, this necessarily occurs in the order of the writing operations performed therein and the predetermined values stored in the memory part concerned are therefore modified by writing or "overwriting" from a certain number of overflow writing operations.

Note that the invention is effective whether the integrity of the bytes dedicated to overflow detection is verified at the beginning of the execution of a task or at the end of the cycle of execution of a task.

According to particular features of the invention, said program is a multitasking program, each task is associated with a stack and on each change of context the

reading and verification operations are effected:

- on the memory part associated with the stack of the task that is going to be executed, or

- on the memory part associated with the stack which, in the order of writing the stacks, is after the stack of the task whose execution has just been interrupted.

Thanks to each of the above features, the method according to the invention applies particularly effectively to multitasking programs.

According to specific features, during the allocation operation, the memory part associated with a stack is adjacent thereto.

Thanks to these features, the slightest stack overflow is detected.

According to specific features, during the allocation operation, the memory part associated with a stack is adjacent to the next stack in the order of writing the stack with which said memory part is associated.

Thanks to these features, only a stack overflow that could disturb the next stack is detected, which provides greater flexibility in managing the monitored stack.

According to specific features, the reading and verification operations are effected by a routine of said program.

Thanks to these features, it is a particularly simple matter to implement the invention.

According to specific features, said predetermined values are all the same.

Thanks to these features, the verification operation is particularly simple because it consists of comparing each value read with the predetermined value.

According to specific features, said predetermined values are different from all the values taken by computer

codes of said program or from all values used in the stacks.

Thanks to these features, it is impossible for the stack overflow to result in the writing of a predetermined value into the read memory part.

According to specific features, when it is found, during the verification operation, that at least one read value has been modified, during a program modification operation, the execution of each program part relating to the stack associated with the memory part that has been read is suspended.

According to other specific features, when it is found, during the verification operation, that at least one read value has been modified, during a program modification operation, the execution of each program part relating to the stack which follows the stack associated with the read memory part in the order of writing the stacks is suspended.

Thanks to each of these features, the consequences of the stack overflow are limited.

A second aspect of the present invention provides a device for detecting overflow of a stack, i.e. of a memory space reserved for a part of a computer program, characterized in that it includes processor means adapted:

- for each stack monitored, to allocate predetermined values to a memory part which is after said stack in the order of writing said stack, and
- on each execution of a program part associated with said stack, to read values present in said memory part and to verify read values.

The invention also provides a franking machine characterized in that it includes a device as succinctly described hereinabove.

The invention also provides:

- means for storing information readable by a

computer or a microprocessor storing instructions of a computer program, characterized in that it enables to implement the method according to the invention as succinctly described hereinabove, and

- partly or completely removable means for storing information readable by a computer or a microprocessor storing instructions of a computer program, characterized in that it enables to implement the method according to the invention as succinctly described hereinabove.

The above device, the above franking machine and the above storage means have the same advantages as the method succinctly described hereinabove, which are not described again here.

Other advantages, objects and features of the invention will emerge from the following description, which is given with reference to the accompanying drawings, in which:

- figure 1 shows a franking machine implementing a stack overflow detection device and method according to the present invention,

- figure 2 is a diagram showing an electronic circuit incorporated in the franking machine shown in figure 1,

- figures 3A and 3B show memory organizations in accordance with the present invention, respectively before and after the detection of a stack overflow, and

- figure 4 is a flowchart of the operation of the electronic circuit shown in figure 2.

The franking machine 1 shown in the drawings includes a device for printing a franking mark and an optional destination address of the envelope on a flat object such as a letter 2.

To print the franking mark in the standardized place provided for this purpose, the letter 2 must be passed through a corridor 5 in the machine 1 which is

delimited by members fastened to the frame, respectively a sliding support 6 which forms the ceiling of the corridor 5, a table 7 which forms its floor, and a ramp which forms one of its lateral limits. The corridor is open at the end opposite the ramp.

To insert the letter 2 into the corridor 5 the letter is placed on the part of the table 7 which projects on the insertion side (the side seen on the left in figure 1), after which the letter is inserted into the corridor 5, as shown in figure 1, until it is driven by means provided for this purpose in the machine 1. The franking mark is printed automatically while the letter 2 is driven along the corridor 5, the franked letter being expelled from the machine at the other end of the corridor 5 (the end seen on the right in figure 1).

For driving the letter 2, the machine 1 includes two rollers 9 and 10, each passing through an opening in the table 7, and respective pressure rollers 12 and 13 for the rollers 9 and 10, each passing through an opening in the support 6.

The rollers 9 and 10 are mounted so that they can rotate relative to the frame of the machine 1 through suspension means 14 shown diagrammatically in figure 1.

The pressure rollers 12 and 13 are mounted on the frame of the machine 1 so that they can rotate but are not suspended from the frame. An electric motor, not shown, is used to drive synchronous rotation of the pressure rollers 12 and 13, for example by means of a belt (not shown) running around three pulleys respectively carried by the motor, the pressure roller 12 and the pressure roller 13.

Because the suspension means 14 urge the rollers 9 and 10 toward the support 6, and therefore toward the pressure rollers 12 and 13, the rollers 9 and 10 are driven by friction against the pressure rollers 12 and 13, either directly or through an object passing through the machine

1, such as the letter 2.

When the letter 2 is inserted into the corridor 5 in the manner shown in figure 1, it eventually encounters the roller 9 and then the pressure roller 12, which drives it in the direction indicated in figure 1 by the horizontal arrow oriented from left to right. At the same time, the roller 9 is lowered as the letter 2 is inserted between the rollers 9 and 12. The letter 2 therefore moves forward in the machine 1 with its face 4 to be printed pressed against and sliding along the surface 17 of the sliding support 6.

The machine 1 includes printing means 19, shown quite diagrammatically in figure 1, for printing the franking mark in its corresponding standardized place and/or the destination address in its corresponding standardized place.

Generally speaking, the printing means 19 apply the franking mark while the letter 2 or the object to be franked is traveling through the machine 1 with its face to be printed pressed against the surface 17 of the sliding support 6, the printing means 19 being located between the pressure rollers 12 and 13.

In the example shown, the printing means 19 are mounted directly on the frame of the machine and are therefore fixed relative to the sliding support 6.

In order for the printing means 19 to be controlled synchronously with forward movement of the object in the machine, a sensor (not shown) is provided to detect the presence of the object and triggers a printing process that is then executed automatically.

To be more precise, a first sensor causes the motor (not shown) to be started when an object begins to be inserted into the machine 1 and a second sensor (not shown) starts the printing process when the object has reached a predetermined location.

Figure 2 shows an electronic control circuit of the

system shown in figure 1. The circuit 100 is shown in the form of a block diagram. It includes, connected by an address and data bus 102:

- a central processing unit 106,
- a random access memory (RAM) 104,
- a flash programmable read-only memory (PROM) 105,
- an input/output port 103 for receiving:
  - the weight of the postal object to be franked, and
  - detection of the postal object by each of the sensors (not shown in the drawings),

and for transmitting:

- motor control signals,

and, independently of the bus 102:

- stepper motors 109,
- presence detection sensors 110,
- a display screen 108 connected to the input/output port 103,
- scales 112 connected to the input/output port 103 and supplying bytes representing the weight of a postal object,
- a keypad 101 connected to the input/output port 103 and supplying bytes representing successively pressed keys of the keypad.

Each of the components shown in figure 2 is well known to the person skilled in the art of microprocessor circuits and, more generally, information processing systems. Those components are therefore not described here.

The random-access memory 104 stores data, variables and intermediate processing results in memory registers which, in the remainder of the description, carry the same name as the data whose value they store. The random-access memory 104 includes in particular registers storing information representing the weight of the postal object to

be franked, the format of the postal object currently being processed, the number of postal objects in the batch currently being processed, up-counter and down-counter values that correspond to franking amounts already applied and remaining to be applied before recharging the machine. The latter registers employ techniques that are known in the franking machine art (during each franking operation, if the down-counter amount is greater than the amount of the franking mark to be applied, it is decremented by the amount of that mark and the up-counter is incremented by the same amount).

The read-only memory 105 is adapted to store the operating program of the central processing unit 106, in a register labeled *"program1"* and the data needed for the program to execute.

The memory 105 referred to as a "random-access memory" is in fact a rewriteable non-volatile memory (i.e. it is not erased when the system is turned off). It can be rewritten only by authorized personnel using secure procedures, so that for the everyday user it is just like a read-only memory.

The central processing unit 106 is adapted to implement the flowchart shown in figure 4 and to organize the random access memory 104 in accordance with figure 3A.

The software (program) of the franking machine is multitasking software, which implies allocation by the processor of a memory space (stack) associated with each task in the random access memory 104.

In the embodiment described and shown, the memory spaces allocated to all the stacks are alternately juxtaposed with memory parts dedicated to detecting or monitoring stack overflow.

The following table shows, in decreasing memory address order, all of the stacks employed by the program, according to the prior art:

stack of task *n*
stack of task *n-1*
.
.
.
stack of task 1
stack of task 0
stack of clock task
stack of background task

Note that the stack pointers move vertically downwards when stacking, reading or writing in the stacks.

It can be easily understood that, if a stack overflows, i.e. if a task writes outside the stack allocated to it, another stack is disturbed (data therein is modified) and the whole of the operation of the franking machine is disturbed.

In the case of franking machines, values stored in the stacks represent "sensitive" values, such as sums of money. It is therefore essential to guarantee that the stacks cannot be violated.

In accordance with the present invention, when the program of the application is started, all of the bytes of each stack are allocated a predefined value. The predefined hexadecimal value A5 is chosen because there is no code resident at the address A5A5.

The stack start address is then fixed for each task of the application. In the embodiment described and shown, the first four bytes of each stack are reserved for overflow control.

The following table, corresponding to figure 3A, shows, in decreasing memory address order, all of the stacks used by the program and memory parts intended to monitor their overflow, in accordance with the invention:

4 bytes associated with stack of task *n* : A5 A5 A5 A5

stack of task *n*

4 bytes associated with stack of task *n*-1 : A5 A5 A5 A5

stack of task *n*-1

4 bytes associated with stack of task *n*-2 : A5 A5 A5 A5

.

.

.

4 bytes associated with stack of task 1 : A5 A5 A5 A5

stack of task 1

4 bytes associated with stack of task 0 : A5 A5 A5 A5

stack of task 0

4 bytes associated with stack of clock task : A5 A5 A5 A5

stack of clock task

4 bytes associated with stack of background task : A5 A5 A5 A5

stack of background task

The stacks 300, 302, 304, 306, 310 and 312 are associated with memory parts 301, 303, 305, 307, 309, 311 and 313 storing predetermined values.

A change of context is an action that interrupts the execution of a task in order to activate the execution of another task. This operation is effected by the kernel of the multitasking program. The functions of the kernel execute a specific routine (the so-called "Hook" routine, which is not internal to the kernel) during a change of context.

The "Hook" routine is a portion of code that is invoked by the kernel during a change of context. When the kernel invokes the "Hook" routine, the current task is the new task. In other words, the current context is the context of the new task (the current stack is the stack of the new task).

In the context of the invention, the above routine is dedicated to checking the integrity of the stack of the new task (i.e. the task which is active after the change of context). The routine verifies the value of the control bytes of the stack of the new task.

In the embodiment described and shown, detecting a stack overflow consists of verifying that the four bytes which precede the stack linked to the new task still contain the values written therein (A5A5A5A5). An overflow of the stack above the stack associated with the new task is detected when not all of the bytes verified have the predetermined value that was written therein.

For example, the following table, corresponding to figure 3B, shows, in decreasing memory address order, all of the stacks used by the program and memory parts which are intended for monitoring their overflow, when the stack *n* 312 has overflowed to the stack *n*-1 310, causing the writing of the hexadecimal codes 34 and 5F for the first two of the four bytes of the memory part 311 which is associated with the stack *n*-1 310:

4 bytes associated with stack of task *n* : A5 A5 A5 A5

stack of task n

4 bytes associated with stack of task *n*-1 : 34 5F A5 A5

stack of task n-1

4 bytes associated with stack of task *n*-2 : A5 A5 A5 A5

.
.
.
4 bytes associated with stack of task 1 : A5 A5 A5 A5
stack of task 1
4 bytes associated with stack of task 0 : A5 A5 A5 A5
stack of task 0
4 bytes associated with stack of clock task : A5 A5 A5 A5
stack of clock task
4 bytes associated with stack of background task : A5 A5 A5 A5
stack of background task

During an initialization operation 401, the central unit 106 initializes values stored in the random access memory 104.

During an operation 402, the central unit 106 initializes the whole of the memory space intended for the stacks and the associated memory parts by placing the hexadecimal value A5 therein.

Then, during an operation 403, execution of the multitasking program is launched by executing a first task and starting a task sequencer.

Thereafter, each time a change of context 404 begins, a Hook routine 405 is executed. The routine 405 includes, in succession:

- an operation 406 to back up the registers of the application on entering the routine (the preceding task, i.e. the one that was active before the beginning of the change of context),

- an operation 407 to read the position of the beginning of the memory part associated with the stack of

the new task (the one that will be active at the end of the change of context),

- an operation 408 to read the bytes placed at the first four addresses, in decreasing order and starting from the address obtained in operation 407,

- a test 409 during which the central processor 106 determines whether each of the bytes read has the hexadecimal value A5 or not,

- if the result of the test 409 is negative, an operation 410 to stop the application and trigger an alarm, after which the program waits for the franking machine to be switched off (413), and

- an operation 411 to restore the registers of the application on quitting the routine.

At the end of the routine 405, the new task executes (operation 412) until the next change of context.

In an alternative embodiment, not shown, at the end of execution of a task the value of the bytes of the memory part associated with the next stack is verified to verify that the stack of the task that has just been executed has not overflowed to the stack that follows it.

In a further alternative embodiment, not shown, the memory part intended for detecting the overflow of a stack is adjacent to the next stack in the order of writing the stack with which said memory part is associated, but is not adjacent to the monitored stack.

In a further alternative embodiment, not shown, the memory part intended for detecting the overflow of a stack is adjacent to the monitored stack, but is not adjacent to the next stack in the order of writing the stack with which said memory part is associated.

CLAIMS

1. A method of detecting overflow of at least one stack (300, 302, 304, 306, 310, 312), i.e. of a memory space reserved for a part of a computer program, characterized in that it includes:

- for each stack monitored, an operation (402) of allocating predetermined values to a memory part (301, 303, 305, 307, 309, 311, 313) which is after said stack in the order of writing said stack, and

- on each execution of a program part associated with said stack (404), an operation (407, 408) to read values present in said memory part and an operation (409) to verify read values.

2. A method according to claim 1, characterized in that said program is a multitasking program, each task is associated with a stack and in that on each change of context (404), the reading and verification operations are effected on the memory part associated with the stack of the task that is going to be executed.

3. A method according to claim 1, characterized in that said program is a multitasking program, each task is associated with a stack and in that on each change of context (404), the reading and verification operations are effected on the memory part associated with the stack which, in the order of writing the stacks, is after the stack of the task whose execution has just been interrupted.

4. A method according to any of claims 1 to 3, characterized in that, during the allocation operation (402), the memory part (305) associated with a stack (304) is adjacent to it.

5. A method according to any of claims 1 to 4, characterized in that, during the allocation operation (402), the memory part (305) associated with a stack (304)

is adjacent to the next stack (306) in the order of writing the stack with which said memory part is associated.

6. A method according to any of claims 1 to 5, characterized in that the reading and verification operations (407 to 409) are effected by a routine (405) of said program.

7. A method according to any of claims 1 to 6, characterized in that said predetermined values are all the same.

8. A method according to any of claims 1 to 7, characterized in that said predetermined values are different from all the values taken by computer codes of said program or from all values used in the stacks.

9. A method according to any of claims 1 to 8, characterized in that when it is found, during the verification operation (409), that at least one read value has been modified, during a program modification operation (410), the execution of each program part relating to the stack associated with the memory part that has been read is suspended.

10. A method according to any of claims 1 to 9, characterized in that when it is found, during the verification operation (409), that at least one read value has been modified, during a program modification operation (410), the execution of each program part relating to the stack which follows the stack associated with the read memory part in the order of writing the stacks is suspended.

11. A device (10) for detecting overflow of at least one stack (300, 302, 304, 306, 310, 312), i.e. of a memory space reserved for a part of a computer program, characterized in that it includes processing means (106) adapted:

- for each stack monitored, to allocate predetermined values to a memory part (301, 303, 305, 307,

309, 311, 313) which is after said stack in the order of writing said stack, and

- on each execution of a program part associated with said stack, to read values present in said memory part and to verify read values.

12. A device according to claim 11, characterized in that the processing means (106) are adapted to execute a multitasking program, each task of which is associated with a stack and, on each change of context (404), to read and verify values of the memory part associated with the stack of the task that is going to be executed.

13. A device according to claim 11, characterized in that the processing means (106) are adapted, on the one hand, to execute a multitasking program each task of which is associated with a stack and, on the other hand, on each change of context (404), to read and verify values of the memory part associated with the stack which is after the stack of the task whose execution has just been interrupted in the order of writing the stacks.

14. A device according to any of claims 11 to 13, characterized in that the processing means (106) are adapted to allocate to a stack (304) a memory part (305) which is adjacent to it.

15. A device according to any of claims 11 to 14, characterized in that the processing means (106) are adapted to allocate to a stack (304) a memory part (305) which is adjacent to the stack (306) after it in the order of writing the stack with which said memory part is associated.

16. A device according to any of claims 11 to 15, characterized in that the processing means (106) are adapted to read and verify values in the memory part by executing a routine of said program (405).

17. A device according to any of claims 11 to 16, characterized in that the processing means (106) are

adapted to process predetermined values that are all the same.

18. A device according to any of claims 11 to 17, characterized in that the processing means (106) are adapted to process predetermined values that are different from all values taken by computer codes of said program or from all values used in the stacks.

19. A device according to any of claims 11 to 18, characterized in that the processing means (106) are adapted, if they determine that at least one read value has been modified, to modify the execution of the program, execution of each program part relating to the stack associated with the memory part that has been read being suspended (410).

20. A device according to any of claims 11 to 19, characterized in that the processing means (106) are adapted, if they have determined that at least one read value has been modified, to modify the execution of the program, execution of each program part relating to the stack following the stack associated with the read memory part in the order of writing the stacks being suspended (410).

21. A franking machine (1), characterized in that it includes a device according to any of claims 11 to 20.

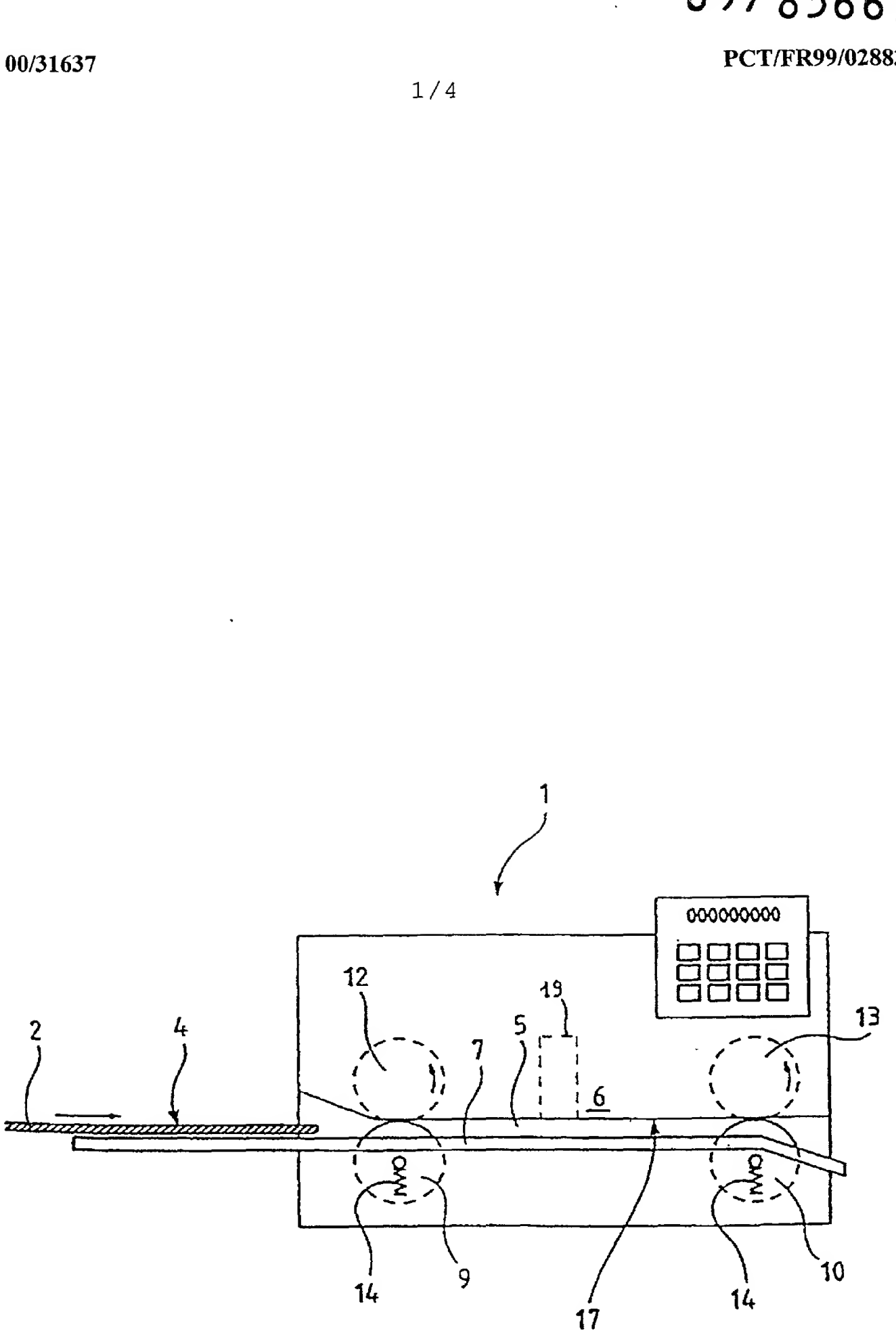

Fig. 1

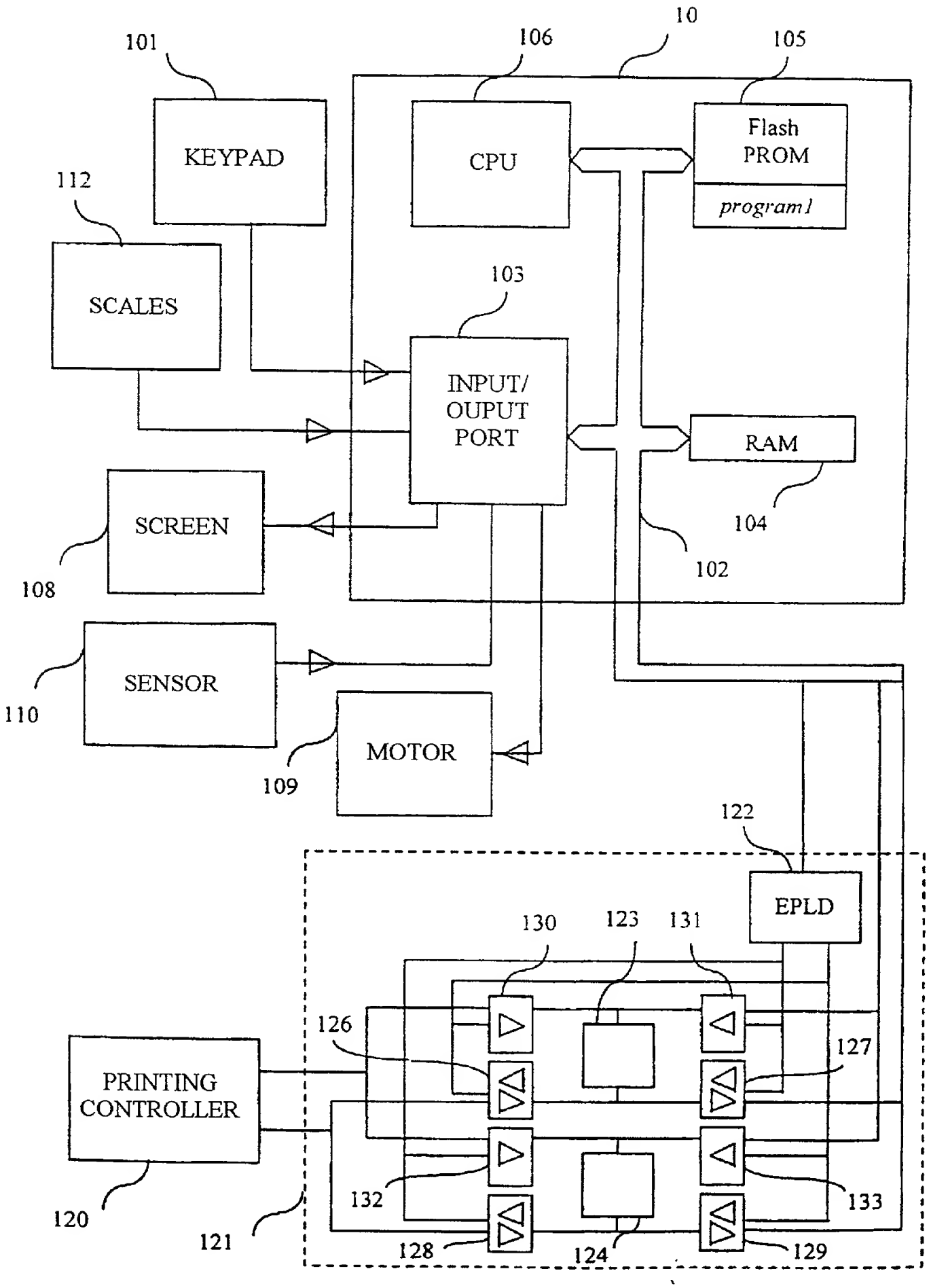
10
101
106
105
KEYPAD
CPU
Flash
PROM
program1
112
SCALES
103
INPUT/
OUPUT
PORT
RAM
104
SCREEN
102
108
SENSOR
110
MOTOR
109
122
EPLD
130
123
131
126
127
PRINTING
CONTROLLER
132
133
120
121
128
124
129

Fig. 2

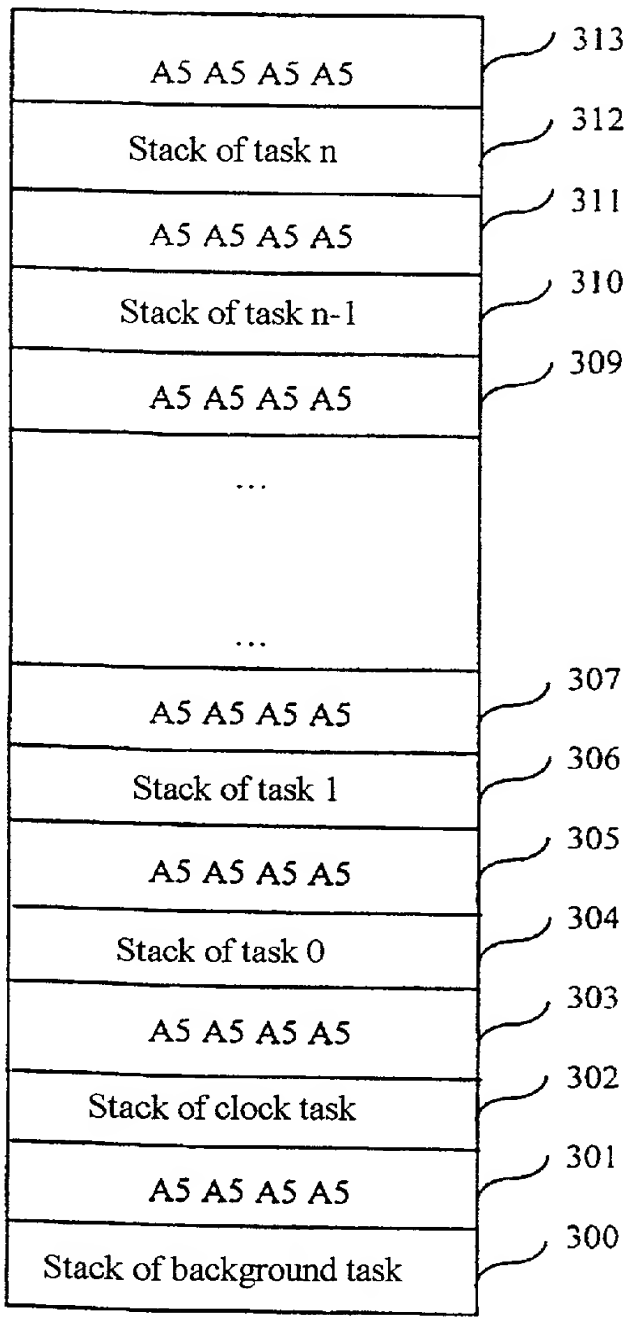
A5 A5 A5 A5
313
Stack of task n
312
A5 A5 A5 A5
311
Stack of task n-1
310
A5 A5 A5 A5
309
...
...
A5 A5 A5 A5
307
Stack of task 1
306
A5 A5 A5 A5
305
Stack of task 0
304
A5 A5 A5 A5
303
Stack of clock task
302
A5 A5 A5 A5
301
Stack of background task
300

Fig. 3A

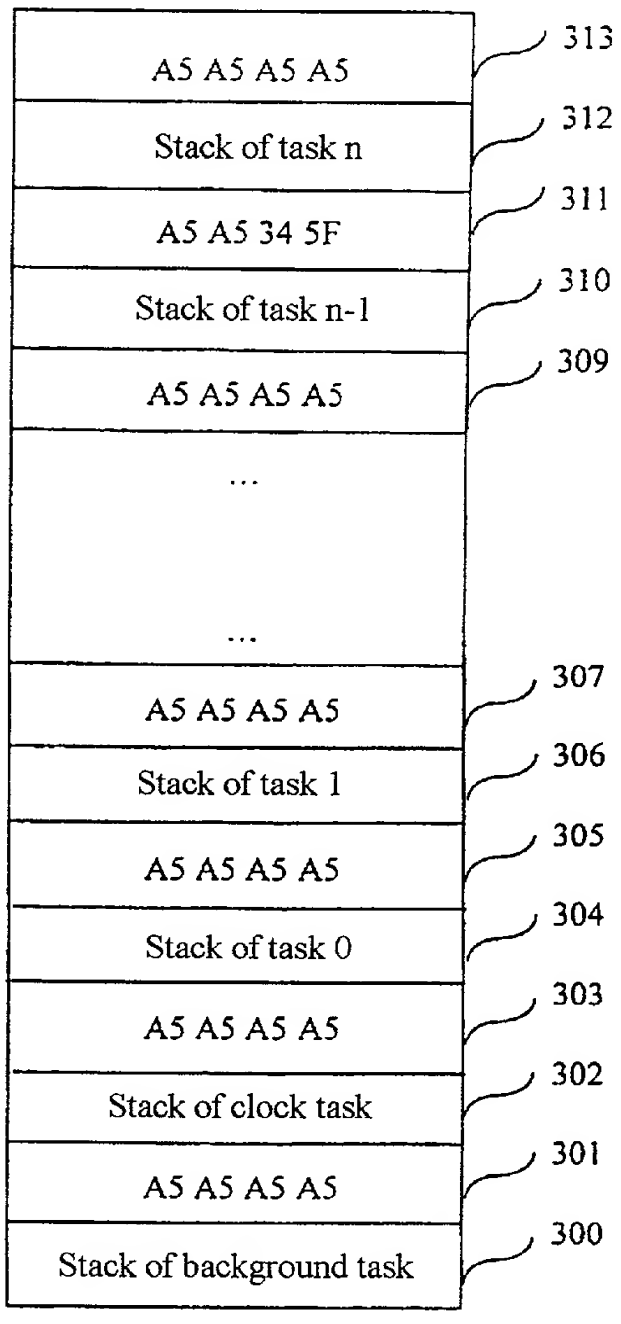
A5 A5 A5 A5
313
Stack of task n
312
A5 A5 34 5F
311
Stack of task n-1
310
A5 A5 A5 A5
309
...
...
A5 A5 A5 A5
307
Stack of task 1
306
A5 A5 A5 A5
305
Stack of task 0
304
A5 A5 A5 A5
303
Stack of clock task
302
A5 A5 A5 A5
301
Stack of background task
300

Fig. 3B

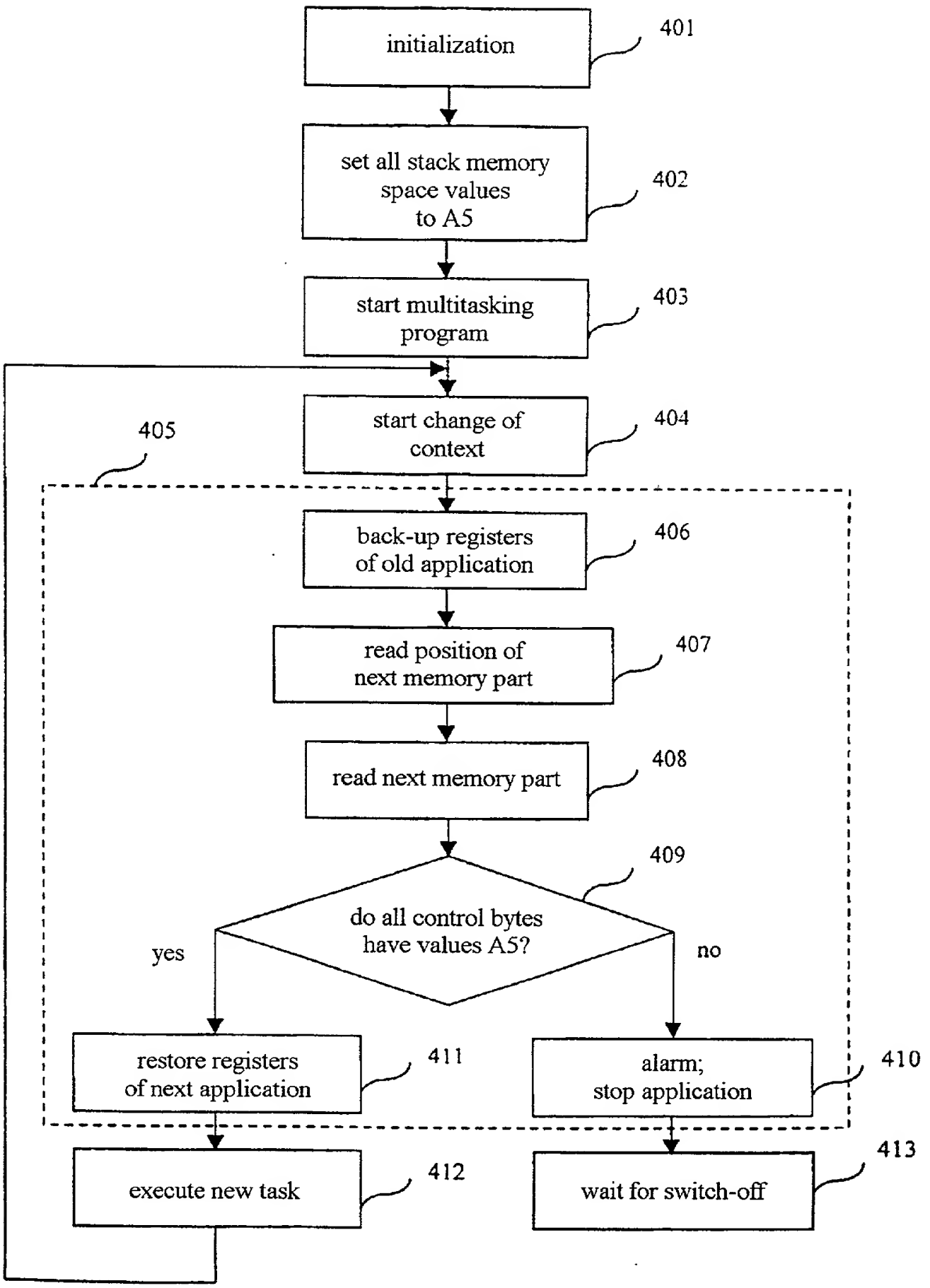
initialization
401
set all stack memory space values to A5
402
start multitasking program
403
start change of context
404
405
back-up registers of old application
406
read position of next memory part
407
read next memory part
408
409
do all control bytes have values A5?
yes
no
restore registers of next application
411
alarm; stop application
410
execute new task
412
wait for switch-off
413

Fig. 4

PTO/SB/105 (2-98)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Declaration and Power of Attorney for Patent Application

Déclaration et Pouvoirs pour Demande de Brevet

French Language Declaration

En tant que l'inventeur nommé ci-après, je déclare par le présent acte que:

As a below named inventor, I hereby declare that.

Mon domicile, mon adresse postale et ma nationalité sont ceux figurant ci-dessous à côté de mon nom.

My residence, post office address and citizenship are as stated next to my name

Je crois être le premier inventeur original et unique (si un seul nom est mentionné ci-dessous), ou l'un des premiers co-inventeurs originaux (si plusieurs noms sont mentionnés ci-dessous) de l'objet revendiqué, pour lequel une demande de brevet a été déposée concernant l'invention intitulée

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

A device for an method of detecting stack overflow in a memory and a franking machine employing them.

et dont la description est fournie ci-joint à moins que la case suivante n'ait été cochée:

the specification of which is attached hereto unless the following box is checked:

☐ a été déposée le _____
sous le numéro de demande des Etats-Unis ou le numéro de demande international PCT
_____ et modifiée le
_____ (le cas échéant).

☒ was filed on NOVEMBER 23, 1999
as United States Application Number or PCT International Application Number
PCT/FR99/02883 and was amended on
_____ (if applicable).

Je déclare par le présent acte avoir passé en revue et compris le contenu de la description ci-dessus, revendications comprises, telles que modifiées par toute modification dont il aura été fait référence ci-dessus.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

Je reconnais devoir divulguer toute information pertinente à la brevetabilité, comme défini dans le Titre 37, § 1.56 du Code fédéral des réglementations.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1 56.

PTO/SB/105 (2-98)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

French Language Declaration

Je revendique par le présent acte avoir la priorité étrangère, en vertu du Titre 35, § 119(a)-(d) ou § 365(b) du Code des Etats-Unis, sur toute demande étrangère de brevet ou certificat d'inventeur ou, en vertu du Titre 35, § 365(a) du même Code, sur toute demande internationale PCT désignant au moins un pays autre que les Etats-Unis et figurant ci-dessous et, en cochant la case, j'ai aussi indiqué ci-dessous toute demande étrangère de brevet, tout certificat d'inventeur ou toute demande internationale PCT ayant une date de dépôt précédant celle de la demande à propos de laquelle une priorité est revendiquée.

I hereby claim foreign priority under Title 35, United States Code, § 119(a)-(d) or § 365 (b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below, and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior foreign application(s) Demande(s) de brevet antérieure(s) (Number) (Numéro)	(Country) (Pays)	Priority Claimed Droit de priorité revendiqué (Day/Month/Year Filed) (Jour/Mois/Année de dépôt)
9814775	FRANCE	NOVEMBER 24, 1998 X
		☐

Je revendique par le présent acte tout bénéfice, en vertu du Titre 35, § 119(e) du Code des Etats-Unis, de toute demande de brevet provisoire effectuée aux Etats-Unis et figurant ci-dessous.

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below.

(Application No.) (N° de demande)	(Filing Date) (Date de dépôt)

Je revendique par le présent acte tout bénéfice, en vertu du Titre 35, § 120 du Code des Etats-Unis, de toute demande de brevet effectuée aux Etats-Unis, ou en vertu du Titre 35, § 365(c) du même Code, de toute demande internationale PCT désignant les Etats-Unis et figurant ci-dessous et, dans la mesure où l'objet de chacune des revendications de cette demande de brevet n'est pas divulgué dans la demande antérieure américaine ou internationale PCT, en vertu des dispositions du premier paragraphe du Titre 35, § 112 du Code des Etats-Unis, je reconnais devoir divulguer toute information pertinente à la brevetabilité, comme défini dans le Titre 37, § 1.56 du Code fédéral des réglementations, dont j'ai pu disposer entre la date de dépôt de la demande antérieure et la date de dépôt de la demande nationale ou internationale PCT de la présente demande:

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or § 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application.

(Application No.) (N° de demande)	(Filing Date) (Date de dépôt)	(Status) (patented, pending, abandoned) (Statut) (breveté, en cours d'examen, abandonné)

Je déclare par le présent acte que toute déclaration ci-incluse est, à ma connaissance, véridique et que toute déclaration formulée à partir de renseignements ou de suppositions est tenue pour véridique; et de plus, que toutes ces déclarations ont été formulées en sachant que toute fausse déclaration volontaire ou son équivalent est passible d'une amende ou d'une incarcération, ou des deux, en vertu de la Section 1001 du Titre 18 du Code des Etats-Unis, et que de telles déclarations volontairement fausses risquent de compromettre la validité de la demande de brevet ou du brevet délivré à partir de celle-ci.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

PTO/SB/105 (2-98)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

French Language Declaration

POUVOIRS: En tant que l'inventeur cité, je désigne par la présente l'(les) avocat(s) et/ou agent(s) suivant(s) pour qu'ils poursuive(nt) la procédure de cette demande de brevet et traite(nt) toute affaire s'y rapportant avec l'Office des brevets et des marques. *(mentionner le nom et le numéro d'enregistrement)*

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

Martin Fleit (Reg. No 16,900), RichardR. Diefendorf (Reg. No. 32,390), Herbert I. Cantor (Reg. No. 24,392), James F. McKeown (Reg No. 25,406), Donald D. Evenson (Reg No. 26, 160), Joseph D. Evans (Reg. No. 26,269), Gary R. Edwards (Reg. No. 31,824), Jeffrey D. Sanok (Reg No. 32,169), Corinne M. Pouliquen (Reg. No. 35,753), David J. Kulik (Reg. No. 36,576) and Paul A. Schnose (Reg. No. 39,361).

Adresser toute correspondance à / Send Correspondence to.

Evenson, McKeown, Edwards & Lenahan, P.L.L.C.
1200 G Street, N.W., Suite 700
Washington, DC 20005-3814

Adresser tout appel téléphonique à *(nom et numéro de téléphone)* / Direct Telephone Calls to. *(name and telephone number)*

Telephone : (202)628-8800
Facsimile : (202)628-8844

Nom complet de l'unique ou premier inventeur 1-00	Full name of sole or first inventor Jean-Marc DERY
Signature de l'inventeur Date	Inventor's signature Date 03/05/2001
Domicile	Residence 2, rue Liouville, 92600 ASNIERES, France
Nationalité	Citizenship French FRX
Adresse postale	Post Office Address 92600 ASNIERES, France
Nom complet du second co-inventeur, le cas échéant	Full name of second joint inventor, if any
Signature du second inventeur Date 2-00	Frédéric L'HOTE Second Inventor's signature Date 3/5/01
Domicile	Residence 5, square Jean Thébaud, 75015 PARIS, France
Nationalité	Citizenship French FRX
Adresse postale	Post Office Address 75015 PARIS, France

(Fournir les mêmes renseignements et la signature de tout co-inventeur supplémentaire.)

(Supply similar information and signature for third and subsequent joint inventors.)